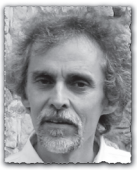
GNU Octave & Octave Forge



Logiciel libre de calcul numérique, visualisation et programmation, compatible avec MATLAB

Jean-Daniel.Bonjour@epfl.ch, EPFL – ENAC-IT et ENAC-SSIE

Introduction

L'évolution des outils informatiques a profondément influencé les méthodes de travail des ingénieurs et chercheurs ainsi que l'enseignement: le traitement numérique des données et leur visualisation, ainsi que les techniques de modélisation et de simulation, se sont notamment généralisés. Dans ce domaine, un logiciel commercial est devenu, ces dernières années, presque incontournable: MATLAB, de la société *The MathWorks*[1]. Son succès s'explique en particulier par une prise en main facile, sa grande richesse fonctionnelle (nombreuses *toolboxes*), des types de données et fonctions *vectorisés*, et son langage interprété de haut niveau permettant à l'utilisateur moyen d'élaborer aisément des applications sans recourir aux langages de programmation classiques (Fortran, C, Java,… désormais réservés aux développeurs professionnels!).

Parallèlement à cette évolution, le mouvement en faveur des **logiciels libres** a pris une grande ampleur. Il s'agit de développements réalisés par la communauté Internet dans le but de se libérer des logiciels commerciaux et proposer des applications répondant avant tout aux besoins des utilisateurs (plutôt qu'à une logique commerciale) et qui soient ouvertes en terme de formats de données et de code source (garantie de pérennité des données). C'est ainsi que des progiciels analogues à MATLAB ont vu le jour, GNU Octave étant celui qui offre actuellement le meilleur degré de compatibilité par rapport à MATLAB (*clone* MATLAB)[2]. L'intérêt des logiciels libres est particulièrement évident dans l'enseignement avec l'avènement des *laptops* (voir la politique *Poseidon* de l'EPFL) en ceci qu'ils n'occasionnent pas de frais de licences, que les fonctionnalités offertes sont largement suffisantes, et qu'ils sont devenus très confortables d'utilisation et robustes.

Cet article est le premier d'une série de contributions sur Octave à paraître dans ce journal. Il ne s'agit pas d'une introduction au langage MATLAB/Octave[3] mais plutôt d'une présentation des caractéristiques principales d'Octave en référence à MATLAB. D'autres articles décriront les expériences réalisées avec Octave à l'EPFL dans l'enseignement, son usage dans des travaux de recherche, le développement de fonctionnali-

SUITE EN PAGE 7

1 ***http://www.mathworks.com/***

2 Outre Octave, l'un des plus connus est Scilab: ***http://www.scilab.org/***

3 Pour cela, on se référera par exemple au support de cours MATLAB/Octave ***http://enacit1.epfl.ch/cours_matlab/***

SUITE DE LA PREMIÈRE PAGE

tés supplémentaires, l'usage de grapheurs alternatifs (Octaviz, OctPlot), le *packaging* réalisé dans le cadre du projet *Poseidon* (Fedora) et sur d'autres plates-formes (Debian). Toute autre contribution sera la bienvenue !

Historique

L'origine de **MATLAB**[4] remonte à la fin des années 70. Développée en Fortran par *Cleve Moler* (Uni. of New Mexico) et s'appuyant sur les librairies LINPACK et EISPACK, la première version de MATLAB comptait environ 80 fonctions, et le seul type de données supporté était la matrice. La société *The MathWorks* a ensuite été fondée en 1984, et les *releases* du produit se sont succédées: MATLAB version 1 sous MS-DOS (1984), v2 avec support Unix (1986), v3 (1987), v3.5 avec graphiques noir-blanc filaires (1990), v4 avec graphiques 2D/3D couleur (1992), v5 avec debugger, profiler, GUI builder et graphiques avec éclairage et ombrage (1996), v6 (2000), v6.5 avec *just-in-time* compilation et support MacOS (2002), v7 (2004). Au cours de cette évolution, les domaines d'utilisation de MATLAB se sont élargis avec l'apparition de nombreuses *toolboxes* commerciales (statistiques, mathématiques symboliques, traitement d'image, traitement de signaux, acquisition de données et contrôle de processus, simulation,…).

Le développement de Octave, quant à lui, est plus récent[5]. Conçu en 1988 en tant que logiciel spécifique dans le cadre d'un cours sur les réacteurs chimiques (*James B. Rawlings* de l'Uni. of Wisconsin-Madison et *John G. Ekerdt* de l'Uni. of Texas), Octave porte le nom d'un professeur auteur de *textbooks* basés sur ce logiciel (nom qui n'a donc ici rien à voir avec la musique). Repensée dès 1992 par *John W. Eaton* dans le sens d'un environnement de calcul numérique plus général et plus flexible, la version 1.0 de **GNU Octave**[6], développée sous Unix, est sortie en février 1994. Elle fait partie du projet GNU[7] et est distribué sous licence GPL. La version courante porte le numéro 2.1.73 (mars 2006) et la version de développement le numéro 2.9.8 (août 2006).

Parallèlement, et en complément à GNU Octave, le projet communautaire **Octave Forge**[8], ainsi nommé, car hébergé sur la plate-forme *Sourceforge.net*, vise à coordonner et regrouper, au sein d'un *GNU Octave Repository*, le développement de fonctions complémentaires principalement orientées dans le sens d'une richesse fonctionnelle approchant celle de MATLAB avec ses *toolboxes* les plus courantes.

Architecture de Octave

GNU Octave doit donc être vu comme le noyau du système Octave, et *Octave Forge* comme un ensemble cohérent d'extensions améliorant son degré de compatibilité par rapport à MATLAB. Dans la suite de cet article, et par souci de simplification, on désignera sous le terme générique **Octave** l'ensemble **GNU Octave + extensions Octave Forge**.

Développé sous Unix/Linux et dans l'esprit modulaire de ce système d'exploitation, Octave ne présente pas le caractère très monolithique de MATLAB. Au contraire, plutôt que d'intégrer toutes les fonctionnalités dans un seul paquet logiciel, Octave coopère avec d'autres outils complémentaires. Il ne dispose ainsi pas de son propre moteur graphique, mais s'appuie sur d'autres grapheurs (*back-ends* graphiques, tels que Gnuplot, Octaviz, OctPlot, epsTK…). De même, Octave ne dispose pas d'un environnement de développement propre (IDE, *Integrated Development Environment*) mais est ouvert à diverses solutions ou projets allant dans ce sens (Octave Workshop[9] et son successeur Octave/MinGW, Yaog[10], Eclipse[11],…). Octave donne ainsi à l'utilisateur une première impression d'austérité: une fenêtre de commande, une fenêtre graphique, éventuellement un éditeur,… et c'est tout (voir figure 1) !

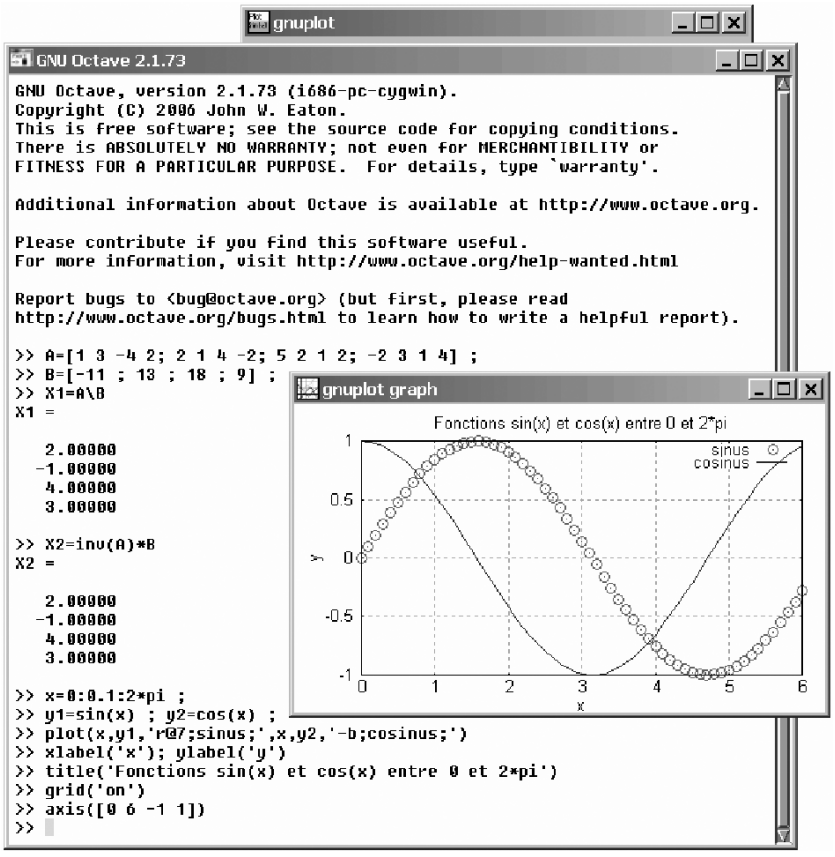


fig. 1 – Fenêtre de commande Octave et fenêtres Gnuplot, ici sous Windows

En raison de ses origines Linux, le portage de Octave sous Windows s'appuie sur **Cygwin**[12] qui est une API d'émulation

4 *http://www.mathworks.com/company/newsletters/news_notes/clevescorner/dec04.html*
5 *http://www.gnu.org/software/octave/about.html*
6 *http://www.gnu.org/software/octave/* , anciennement *http://www.octave.org/*
7 *http://www.gnu.org/*
8 *http://octave.sourceforge.net/*
9 *http://www.math.mcgill.ca/loisel/octave-workshop/*
10 *http://yaog.sourceforge.net/*
11 *http://www.eclipse.org/*
12 *http://www.cygwin.com/*

Linux pour Windows, sous licence GPL et maintenue par la société Red Hat.

Installation sous Linux

Bien qu'il soit possible d'installer Octave en récupérant les codes sources et en les compilant[13], il est plus facile d'utiliser les paquetages binaires proposés par les différentes distributions Linux. Le Wiki Octave décrit les paquetages nécessaires et les étapes d'installation pour les distributions Linux suivantes[14]: *Debian*, *Gentoo*, *Fedora Core*, *Red Hat* et *SuSE*.

À titre d'exemple sous **Fedora Core**, les paquetages relatifs à Octave, qui ne font plus partie de la distribution de base depuis FC4 mais se trouvent dans *Fedora Extra*, sont dénommés `octave` (noyau Octave) et `octave-forge` (extensions Forge). Pour faire usage des fonctions graphiques de base, il faut en outre disposer du paquetage `gnuplot`. Avec la commande `yum install octave-forge`, on installe donc très simplement Octave-Forge ainsi que tous les paquetages dépendants.

Dans le cadre de la distribution Fedora du projet *Poseidon* EPFL, il faut noter que les différents paquetages Octave sont déjà installés (vérifiez-le avec la commande `rpm -qa | grep octave`). L'équipe *Poseidon* a en outre *packagé* et préinstallé `octaviz` qui consiste en un *back-end* graphique alternatif à Gnuplot particulièrement intéressant pour produire de beaux graphiques 3D (voir un prochain article sur ce sujet).

Installation sous Mac OS X

Le Wiki Octave décrit trois possibilités d'installation de Octave sous Mac OS X basées sur des paquetages binaires[15]:

- Fink: voir ***http://fink.sourceforge.net***
- DarwinPorts: voir ***http://www.opendarwin.org/projects/darwinports***
- binaire HPC.sourceforge: voir ***http://hpc.sourceforge.net/#octave.***

Usage de Octave en ligne de commande sous Linux et MacOS X

Sous Linux et MacOS X, Octave s'utilise tout simplement dans une fenêtre terminal. La commande `octave --help` présente les différentes options de lancement. L'une d'entre elles, l'option `--traditional`, sert à positionner certaines variables *builtins* en *mode compatibilité* MATLAB (voir plus bas). Il peut donc être utile d'intégrer cette option dans un *alias* de démarrage Octave.

Pour lancer Octave dans une fenêtre terminal, on commence généralement par se placer, avec une commande `cd`, dans le dossier de travail où se trouvent les données à traiter ainsi que les éventuels scripts et fonctions de l'utilisateur (M-files), puis on passe au *shell* la commande `octave`. Avant de passer la main à l'utilisateur et afficher son *prompt*, Octave exécute d'éventuels *prologues*. Il s'agit de scripts (M-files) de nom `.octaverc` et contenant des instructions d'initialisation Octave/MATLAB librement définies par l'utilisateur. Octave exécute ainsi successivement: le prologue se trouvant dans le répertoire-racine de l'utilisateur, puis celui du répertoire courant (c'est-à-dire le répertoire depuis lequel on a démarré Octave).

On peut également exécuter en batch, directement depuis le *shell*, tout script Octave/MATLAB avec la commande `octave -qf script.m`. Si le script commence par la ligne `#!/usr/bin/octave -qf`, il est même possible de l'exécuter depuis le *shell* en l'invoquant simplement par son nom, par exemple `./script.m` (pour autant qu'il aie été préalablement mis en mode *execute* avec la commande `chmod u+x script.m`).

Installation sous Windows

Comme déjà mentionné, le portage de Octave sous Windows s'appuie sur **Cygwin**. On distingue essentiellement deux techniques d'installation de Octave sous Windows:

- installation d'un paquet **stand-alone** de type *bundle* intégrant: Cygwin, Octave, les extensions Forge et le grapheur Gnuplot;
- via l'**installeur Cygwin**[16], à travers Internet: installation des paquetages précompilés Octave (ou éventuellement des paquetages sources qu'il faut alors compiler soi-même), tel que décrit dans le Wiki Octave[17]; il faudra sélectionner, dans la catégorie *Math*, les paquetages `octave`, `octave-headers` et `octave-forge`, et dans la catégorie *Graphics*, le paquetage `gnuplot`.

La seconde méthode ne devrait être employée que par les utilisateurs expérimentés ou désirant disposer de la plus récente version d'Octave (2.9). Pour l'utilisateur standard, la première méthode est nettement plus simple et rapide, et c'est celle que nous décrivons ci-après. Notez aussi que Octave est déjà préinstallé avec la distribution Windows du projet *Poseidon* EPFL!

Le **paquet stand-alone** Octave pour Windows se présente sous la forme d'un *installeur* téléchargeable depuis Sourceforge[18]. Attention: comme il intègre Cygwin, ce paquet ne doit pas être utilisé si vous avez préalablement installé Cygwin sur votre machine (auquel cas il faut obligatoirement recourir à la seconde méthode d'installation). La version actuelle de ce paquet (mars 2006) est basée sur Octave 2.1.73, et le kit d'installation se nomme `octave-2.1.73-1-inst.exe` (23 MB). Il intègre les composants suivants (une fois installé, Octave occupe env. 140 MB pour 8000 fichiers):

- GNU Octave 2.1.73
- extensions Octave Forge 2006-03-17
- environnement de base Cygwin 1.5.19
- Gnuplot graph 4.0 patchlevel 0, avec *mouse-support*
- éditeur de texte Scintilla SciTE 1.68

Une notice d'installation détaillée est disponible sous ***http://enacit1.epfl.ch/cours_matlab/octave.html***. Une fois l'installation terminée, nous vous conseillons encore

[13] ***http://wiki.octave.org/wiki.pl?BuildFromSourcev***
[14] ***http://wiki.octave.org/wiki.pl?OctaveForGNULinux***
[15] ***http://wiki.octave.org/wiki.pl?OctaveForMac***
[16] ***http://www.cygwin.com/***
[17] ***http://wiki.octave.org/wiki.pl?OctaveForWindows***
[18] ***http://sourceforge.net/project/showfiles.php?group_id=2888***

d'effectuer les réglages suivants pour un meilleur confort d'utilisation:

Script de démarrage Octave

(en principe sous `C:\Program Files\GNU Octave 2.1.73\octave.sh`):

- faites pointer la variable `HOME` sur votre dossier de démarrage Octave, en utilisant la syntaxe `/cygdrive/lettre/dossier` (où `lettre` désigne la lettre de la partition Windows, telle que C, D…);
- intégrez éventuellement le *path* ci-dessus dans la variable `PATH`;
- ajoutez le switch `--traditional` à la suite de la commande de démarrage `octave.exe` (pour une meilleure compatibilité de Octave par rapport à MATLAB).

Raccourci de lancement GNU Octave 2.1.73

(sur le bureau et dans le menu *Démarrer*> *Programmes*> *GNU Octave*): sélectionnez son icône et, avec le bouton de <droite>, choisissez **Propriétés**, puis éditez avec prudence la cible de la façon suivante:

- paramètre `-geometry`: adaptez la taille de fenêtre à vos besoins, p.ex. `100x45` (*colonnes* x *lignes*);
- police de caractère, paramètre `-fn`: nous vous recommandons la police `Fixedsys-9` (plus lisible que `Lucida Console-12`);
- couleur du curseur: c'est le paramètre `-cr color`;
- pour augmenter la taille de la mémoire de *scroll* (p.ex. 1000 lignes), il faudrait ajouter le paramètre `-sl 1000`.

Comme dans sa version Linux, Octave supporte aussi un mécanisme de *prologue* utilisateur. Le fichier nommé `.octaverc` se trouvant dans le dossier-racine de l'utilisateur (défini par la variable `HOME`) sera exécuté automatiquement à chaque lancement de Octave.

Spécificités de Octave par rapport à MATLAB

Après avoir décrit les spécificités de Octave relatives à son installation, sa configuration et ses mécanismes de démarrage et de prologue-utilisateur, nous passons ici en revue d'autres particularités intéressantes de Octave.

Variables builtins

Le comportement de Octave est dicté par un certain nombre de variables spécifiques, dites *builtins*, dont on obtient la liste avec la commande `who -builtins`. Ces variables peuvent être modifiées par l'utilisateur, que ce soit interactivement ou via le *prologue* de démarrage. L'option `--traditional` de démarrage Octave, mentionnée précédemment, positionne certaines de ces variables de façon que Octave se comporte de la manière la plus compatible possible par rapport à MATLAB, en particulier:

- `default_save_format='mat-binary'`: enregistrement des fichiers de *workspace* (`*.mat`) au format binaire MATLAB v5;
- `PS1='>> '`: prompt identique à celui de MATLAB;
- `page_screen_output=0`: défilement de l'affichage non paginé; la commande `more on` positionne cette variable à `1` pour un affichage paginé (défilement d'une ligne avec la touche `<enter>`, d'une page en avant avec `<espace>` ou `<f>`, d'une page en arrière avec `<b>`);
- `beep_on_error=1`: *beeper* en cas d'erreur;
- `fixed_point_format=1`: dans l'affichage de vecteurs ou matrices, peut débuter l'affichage par un facteur de multiplication commun à tous les éléments.

Aide et documentation

Pour obtenir de l'aide sur une fonction, on utilise, comme dans MATLAB, la commande `help` *fonction*. On peut en outre faire une recherche via l'index du manuel Octave avec la commande `help -i` *mot-clé*.

Intégré à certaines distributions Octave (p.ex. le paquet *stand-alone* pour Windows), le manuel d'utilisation des fonctions de base Octave est également disponible sur Internet[19]. S'agissant des extensions Octave Forge, on se référera à ***http://octave.sourceforge.net/index/***.

Copier et coller

Sous Windows, l'implémentation du copier/coller Octave est inhabituelle pour ce système d'exploitation, car elle se base sur le standard X-Window du monde Unix/Linux, à savoir:

- `copier`: la sélection courante est automatiquement copiée dans le *presse-papier*;
- `coller`: le contenu du *presse-papier* est collé à l'aide du bouton du `<milieu>` de la souris.

Auto-completion

A la manière des *shells* Unix, l'usage de la touche `<tab>` permet d'étendre automatiquement un nom de variable, de fonction, de fichier ou de répertoire[20]. Par exemple `auto<tab>` n'affiche rien (car ambiguïté), mais presser une seconde fois `<tab>` affichera la fonction `autocov` et la variable `automatic_replot`. Cependant si l'on frappe `autom<tab>`, cela insérera directement la variable `automatic_replot`, car il n'y a dans ce cas plus d'ambiguïté possible.

Historique des commandes

Comme pour les *shells* Unix, Octave maintient un *historique* des dernières commandes passées (le nombre de commandes mémorisées étant défini par la variable `history_size`). Celui-ci est implémenté par le fichier `.octave_hist` se trouvant dans le dossier racine de l'utilisateur. La commande `history` permet d'afficher les dernières commandes, et la commande `run_history` *numéro* permet de relancer la commande de numéro spécifié.

Fichiers de workspace

Comme cela a été mentionné, il est nécessaire que la variable `default_save_format` soit positionnée à `'mat-binary'` (ce que fait l'option de lancement `--traditional`) si l'on souhaite que les fichiers de *workspace* (MAT-files) générés par Octave soient au format binaire MATLAB v5.

Contrairement à MATLAB, si l'on ne spécifie par l'extension de nom de fichier `.mat` lorsqu'on sauvegarde un *works-*

[19] ***http://www.gnu.org/software/octave/doc/interpreter/***
[20] Fonction apparue, sous MATLAB, après la version 5

pace (commande `save`), le fichier n'aura pas d'extension. Il est donc sage de prendre l'habitude de spécifier cette extension si l'on souhaite un nommage cohérent avec MATLAB. Mais il n'est, bizarrement, pas nécessaire de spécifier cette extension lorsqu'on charge un *workspace* (commande `load`).

Gnuplot, grapheur par défaut de Octave

Le *grapheur par défaut* utilisé par Octave (que ce soit le noyau GNU Octave ou les extensions Octave Forge) est donc **Gnuplot**[21]. Les fonctions graphiques de base bien connues sous MATLAB (fonctions de tracé telles que: `plot`, `plot3`, `text`, `contour`, `mesh`, `meshc`… ou de mise en forme: `axis`, `grid`, `x/y/zlabel`, `legend`, `title`…) font ainsi automatiquement appel à ce *back-end* Gnuplot.

Dans ses versions 3.0 et 4.0, Gnuplot reste hélas essentiellement orienté *dessin de lignes* (traits). Il ne se prête donc pas bien aux types de graphiques dans lesquels des surfaces doivent être coloriées (p.ex. `fill`, `pie`, `bar`, `surf`…); Gnuplot simule alors le remplissage par un hachurage. Cela évoluera, on l'espère, avec Gnuplot 4.2… Mais la bonne nouvelle, c'est que l'ouverture de Octave permet de faire usage d'autres moteurs graphiques. Parmi les divers projets d'interfaçage de grapheurs alternatifs (efforts principalement réalisés sous Linux), ceux qui sortent actuellement du lot sont:

- **Octaviz**[22]: *wrapper* donnant accès aux classes VTK[23] (Visualization ToolKit, librairie 3D); il est intégré à la distribution Octave *Poseidon*-EPFL sous Fedora;
- **Octplot**[24]: grapheur actuellement 2D, ultérieurement 3D;
- **epsTK**[25] (encapsulated postscript toolkit): jeu de fonctions pour créer des graphiques 2D très sophistiqués au format EPS (nécessitant le logiciel libre Ghostscript pour la visualisation à l'écran); il était intégré dans les anciens paquets Octave stand-alone (2.1.42) pour Windows;
- **matplotlib**[26]: librairie de dessin 2D s'appuyant sur Python et offrant un bon degré de compatibilité avec les fonctions graphiques MATLAB;
- **PLplot**[27]: librairie de dessin 2D/3D.

Ces solutions seront décrites dans d'autres articles de ce journal. Nous nous limiterons donc ici à une présentation succincte des possibilités graphiques de Octave en liaison avec Gnuplot.

Limitations de Gnuplot

Outre la difficulté de représenter des surfaces remplies, Octave-Gnuplot n'implémente pas la notion de *handle graphics* connue sous MATLAB. De plus, le mode *multiple-plots* (plusieurs graphiques dans la même fenêtre de figure) ne fonctionne pas toujours bien (problèmes de rafraîchissement de la fenêtre graphique). Finalement, mais sous Windows seulement, Gnuplot ne permet pas de manipuler simultanément plusieurs fenêtres de figures (mode *multiple graphic windows*).

La fenêtre Gnuplot et ses raccourcis

La fenêtre de graphique intitulée *Gnuplot graph* apparaît automatiquement lorsqu'on fait usage de commandes réalisant un tracé (graphique 2D ou 3D). Sous Windows, une fenêtre *Gnuplot* apparaît également, celle-ci étant initialement repliée dans la barre de tâches: il s'agit de la console dans laquelle défilent les *vraies commandes* Gnuplot (découlant des fonctions graphiques Octave) destinées à la fenêtre graphique. Attention: sous Windows toujours, ces deux fenêtres Gnuplot ne doivent pas être refermées manuellement par leur case de fermeture (engendre des blocages lorsque l'on génère d'autres graphiques durant la session Octave), mais plutôt via la commande `closeplot`.

Octave active automatiquement le *mode souris* de Gnuplot. On peut aussi faire cela manuellement avec la touche `<m>` (bascule d'activation/désactivation) dans la fenêtre graphique Gnuplot, ou utiliser les commandes `set mouse` resp. `unset mouse`, dans la fenêtre de commande Gnuplot. Par un <cliquer-glisser> avec la souris dans la fenêtre de graphique:

- pour un graphique 2D: zooms interactifs précis avec `<droite>glisser<droite>` (outre la commande `axis`); pour faire un zoom out, on passera la commande `axis('auto')` ou le raccourci `<a>`
- pour un graphique 3D:
  - `<gauche-glisser>`: rotation 3D;
  - `<milieu-déplac.horizontal>`: zoom in/out (utiliser `<ctrl>` pour graphiques complexes);
  - `<milieu-déplac.vertical>`: changement d'échelle en Z (à combiner avec `<ctrl>` pour des graphiques complexes);
  - `<maj-milieu-déplac.vertical>`: changement d'origine Z (à combiner avec `<ctrl>` pour des graphiques complexes).

En cliquant avec <droite> sur la barre de titre du graphique, un sous-menu Options apparaît comportant notamment les articles suivants:

- `Copy to Clipboard`: copie en format vecteur de la figure dans le presse-papier (en vue de la coller ensuite dans un autre document);
- `Choose Font`: permet de changer la police de caractère des textes;
- `Line Styles`: permet de changer les styles de ligne (par défaut: Border `noir solid 1`, Axis (grille) `gris dot 1`, Line1 `rouge solid 1`, Line2 `vert solid 1`, Line3 `bleu solid 1`,…);
- `Background`: permet de changer la couleur de l'arrière-plan (blanche par défaut);
- `Color` (activé par défaut): en désactivant cette option, le graphique s'affichera en noir-blanc.

Pour autant que l'on dispose de la version *mouse-enabled* de Gnuplot, divers raccourcis-clavier sont possibles dans la fenêtre graphique, en particulier:

- `<g>`: affichage/masquage de la grille (bascule);
- `<l>`: axe Y (2D) ou Z (3D) logarithmique/linéaire (bascule);
- `<L>`: axe se trouvant le plus proche du curseur logarithmique/linéaire (bascule);

---

21 *http://www.gnuplot.info/*
22 *http://octaviz.sourceforge.net/*
23 *http://www.vtk.org/*
24 *http://octplot.sourceforge.net/*
25 *http://www.epstk.de/*
26 *http://matplotlib.sourceforge.net/*
27 *http://plplot.sourceforge.net/*

- <b>: affichage/masquage d'une boîte dans les graphiques 3D (bascule);
- <a>: *autoscaling* des axes (utile après un zoom !);
- <7>: même échelle (ratio) pour les axes X et Y (bascule);
- <p> et <n>: facteur de zoom précédent, resp. suivant;
- <u>: dé-zoomer;
- <e>: équivalent à la commande `replot`.

Exemples Octave-Gnuplot

Nous terminons cette présentation du couple Octave-Gnuplot par quelques exemples de graphiques typiques.

La figure 2, réalisée par le code Octave ci-dessous, montre un exemple de graphique 2D classique. Notez la manière propre à Octave de définir la légende (passée en paramètre dans la fonction `plot`):

```
x=linspace(0,20,30);
y1=sin(x)./exp(x/10);
y2=1.5*sin(2*x)./exp(x/10);
plot(x,y1,'r-@7;amortisseur 1;',…
   x,y2,'-b@8;amortisseur 2;');
grid('on');
xlabel('Duree (sec)');
ylabel('Amplitude (mm)');
text(11,-0.8,'annotation…');
title('Test amortisseurs');
```

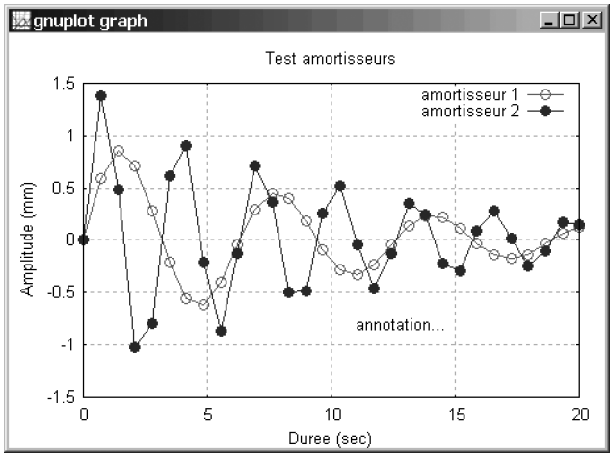


fig. 2 – Graphique 2D classique

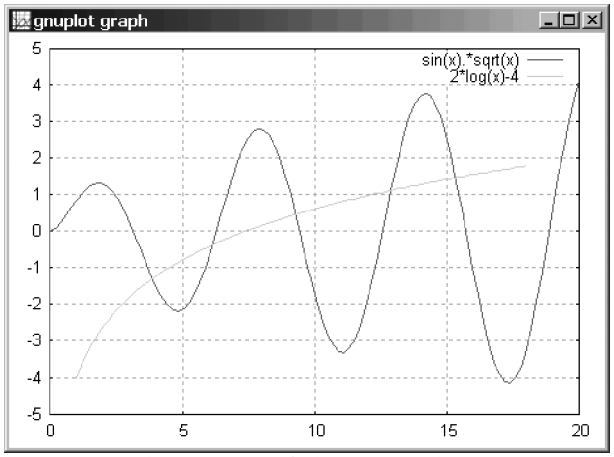


fig. 3 – Tracé de fonctions y=f(x)

Le code ci-dessus et la figure 3 illustrent la technique de tracé de fonctions de type y=f(x):

```
fplot('sin(x).*sqrt(x)',[0 20]);
hold('on');
fplot('2*log(x)-4',[1 18]);
grid('on');
```

Le code suivant, qui génère la figure 4, montre que la fonction `errorbar` offre, sous Octave, davantage de possibilités que sous MATLAB (boîtes d'erreur 2D, et non seulement barres d'erreur). On y voit également l'usage de la fonction `hold('on')` pour superposer deux graphiques.

```
x=[1 3 5 9 13]; y=[3 3.8 3.5 4.1 3.8];
lowX=[0.5 1 1.5 1 2]; upX=[2 1 2 3 2];
lowY=[0.2 0.1 0.3 0.2 0.3]; upY=[0.1 0.2 0.1
0.1 0];
plot(x,y,'r-8');
hold('on');
errorbar(x,y,lowX,upX,lowY,upY,'b#~>');
axis([0 16 2.6 4.4]); grid('on');
```

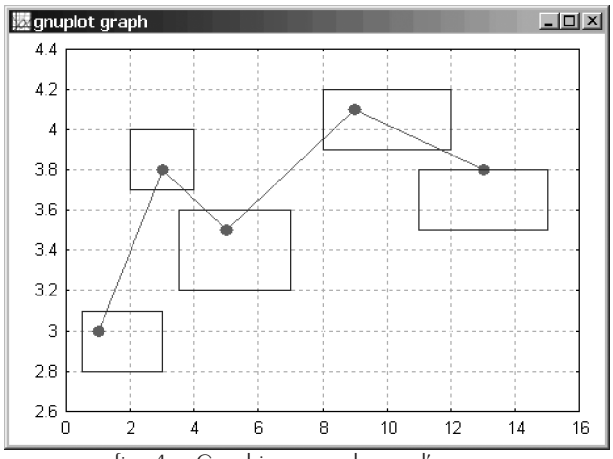


fig. 4 – Graphique avec boxes d'erreurs

Le code ci-dessous, associé à la figure 5, montre comment on peut représenter en 2D des données 3D sous la forme de *contours* (interpolation de courbes de niveau):

```
x=-2:0.1:2;
y=x;
[X,Y]=meshgrid(x,y);
Z=10*sin(X).*sin(Y).*exp(-X.^2 + X.*Y -
Y.^2);
contour(X,Y,Z,20);
```

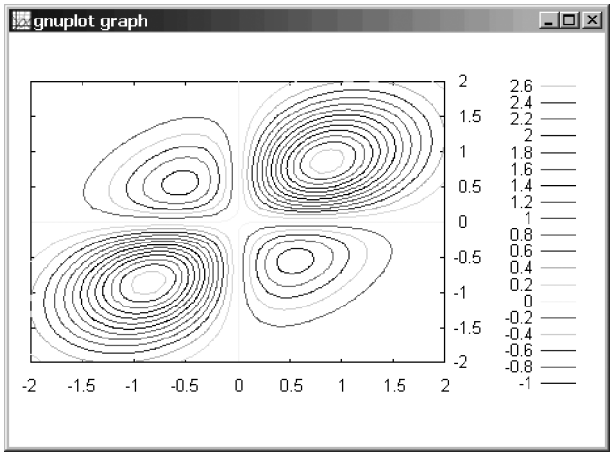


fig. 5 – Graphique 2D de représentation de données 3D en contours

En réutilisant les données de l'exemple précédent, on les représente, dans le graphique 3D de la figure 6, sous forme d'affichage combiné surface+contours, par l'instruction:

```
surfc(X,Y,Z);
```

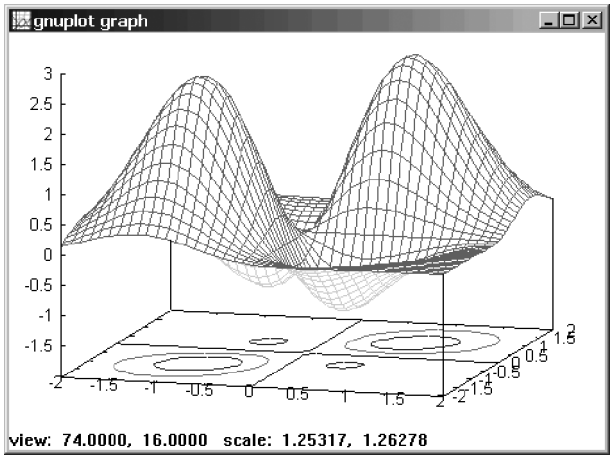


fig. 6 – Représentation d'une surface 3D

Finalement, le code qui suit illustre, dans la figure 7, l'affichage 3D de semis de points et de lignes:

```
z1=0:0.1:10*pi;
x1=z1.*cos(z1);
y1=z1.*sin(z1);
x2=60*rand(1,20)-30; % 20 pts de coord.
y2=60*rand(1,20)-30; % -30 < X,Y < 30
z2=35*rand(1,20);   %  0 < Z < 35
plot3(x1,y1,z1,';x=z*cos(z) y=z*sin(z);',…
   x2,y2,z2,'b8;semis aléatoire;');
grid('on');
xlabel('x'); ylabel('y'); zlabel('z');
```

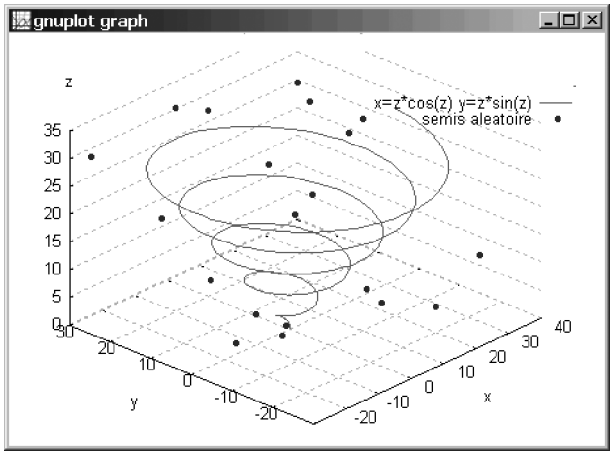


fig. 7 – Affichage 3D de lignes et semis de points

Fonctionnalités méconnues d'Octave (et de MATLAB)

Bien que l'objectif de cet article ne soit pas une introduction au langage MATLAB/Octave, nous présentons ici quelques fonctionnalités, souvent méconnues des usagers coutumiers de MATLAB, qui illustrent la puissance de Octave et son bon niveau de compatibilité par rapport à MATLAB.

Tableaux multidimensionnels

Supportés depuis la version 2.1.51 de Octave, les *tableaux multidimensionnels* (*multidimensional arrays*, *ND-arrays*) sont des matrices de plus de 2 dimensions, c'est-à-dire ayant trois indices ou davantage. Un tableau à 3 dimensions est ainsi un *empilement* de matrices 2D de dimensions identiques (voir figure 8), un tableau à 4 dimensions pourrait être vu comme un vecteur de tableaux 3D de dimensions identiques, etc.

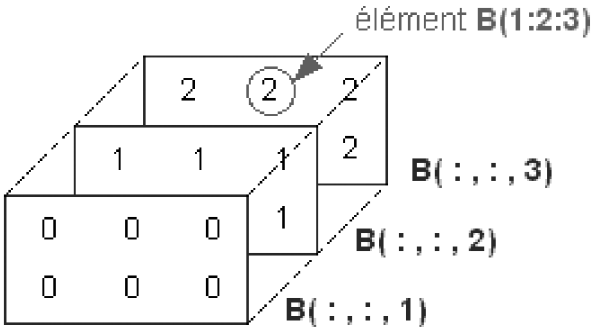


fig. 8 – Exemple d'un tableau 3D de dimension 2x3x3

A titre d'exemple, pour initialiser le tableau de la figure 2 on peut utiliser les instructions suivantes:

- `B(2,3,3)=2` génère le tableau B de dimension 2x3x3 et initialise tous ses éléments à la valeur `2`;
- `B(:,:,2)=[1 1 1; 1 1 1]` réinitialise tous les éléments de la seconde «couche» de ce tableau 3D à la valeur `1`;
- plus simplement, `B(:,:,1)=zeros(2,3)` ou `B(:,:,1)=0` réinitialise tous les éléments de la première «couche» de ce tableau 3D à la valeur `0`.

Il est bien clair que la plupart des fonctions *vectorisées* de Octave/MATLAB permet de manipuler directement de tels tableaux sans devoir implémenter des boucles `for` sur les indices pour accéder aux différents éléments du tableau.

Structures

Une *structure* (*enregistrement*, *record*) est un type d'objet MATLAB/Octave existant aussi dans d'autres langages et se composant de plusieurs *champs* nommés (*fields*) qui peuvent être de types différents (chaînes, matrices, tableaux cellulaires…), champs qui peuvent eux-mêmes se composer de sous-champs, etc. La syntaxe est analogue à celle de langages orientés-objet tel que Ruby. MATLAB/Octave permet, logiquement, de créer des *tableaux de structures* (*structures array*) multidimensionnels.

nom: Dupond **prenom**: Jules
age: 25
code_postal: 1010 **localite**: Lausanne
enfants: -
tel.prive: 021 123 45 67 **tel.prof**: 021 987 65 43

nom: Durand **prenom**: Albertine
age: 30
code_postal: 1205 **localite**: Geneve
enfants: Arnaud Camille
tel.prive: - **tel.prof**: -

fig. 9 – Tableau de structures «personne» à 2 éléments

Le tableau de structures illustré par la figure 9 pourrait être initialisé de la façon suivante:

- avec `personne.nom='Dupond'`, la structure est mise en place et contient le nom de la 1ère personne (vérifiez avec `whos personne`);
- `personne.prenom='Jules'` ajoute le champ `prenom` à cette structure et définit le prénom de la personne;
- et ainsi de suite pour les autres champs: `personne.age=25; personne.code_postal=1010; personne.localite='Lausanne'; personne.tel.prive='021 123 45 67'`, etc.;
- on peut, à ce stade, vérifier le contenu de la structure en frappant: `personne`;
- l'ajout d'une seconde personne à ce tableau de structures se ferait avec: `personne(2).nom='Durand'; personne(2).prenom='Albertine'`, etc.
- pour définir les enfants de cette seconde personne, on peut créer un champ `enfants` de type *tableau cellulaire* (voir plus bas) en définissant les prénoms avec: `personne(2).enfants={'Arnaud','Camille'}`.

On accède ensuite aux éléments de la structure ainsi:

- `personne(1).tel.prive` retourne le No de téléphone privé de la 1ère personne;
- `personne(2).enfants{1}` retourne le nom du 1er enfant de la 2e personne;
- `cell_tab2 = { personne.nom }` retourne l'objet `cell_tab2`, de type *tableau cellulaire*, contenant l'ensemble des noms de tous les individus.

Il est intéressant de noter que Octave est capable de sauvegarder des tableaux de structures sous forme de fichier texte (avec `save -ascii`), ce que MATLAB ne sait pas faire. Cela peut être utile si l'on doit exporter des données de type *structure* vers d'autres applications.

Tableaux cellulaires

Le *tableau cellulaire* (*cells array*) est le type de données MATLAB/Octave le plus polyvalent. Il se distingue du «tableau standard» en ce sens qu'il peut se composer d'éléments de dimensions et types différents (scalaire, vecteur, chaîne, matrice, structure, voire même tableau cellulaire… permettant ainsi de faire des tableaux cellulaires imbriqués dans des tableaux cellulaires). Les tableaux cellulaires sont particulièrement utiles pour stocker du texte. Les tableaux classiques présentent en effet l'inconvénient que toutes les lignes de texte doivent avoir le même nombre de caractères, ce qui implique de compléter les fins de lignes par des blancs (puis de les enlever avec la fonction `deblank`).

'hello'	22 23 24 25
personne nom: 'Dupond' age: 25 nom: 'Durand' age: 30	{ 'quatre' 44 }

fig. 10 – Tableau cellulaire formé de 4 cellules de type différent

Pour définir un tableau cellulaire et accéder au contenu de ses éléments, on recourt aux accolades `{ }`. Celles-ci seront utilisées soit au niveau des indices des éléments du tableau, soit dans la définition de la valeur introduite dans la cellule. Illustrons ces différentes syntaxes par la structure de la figure 4.

Ce tableau serait initialisé par les instructions suivantes:

- avec `T(1,1)={ 'hello' }` ou `T{1,1}='hello'` on définit la première cellule (notez bien l'usage des parenthèses et des accolades !);
- avec `T(1,2)={ [22 23; 24 25] }` ou `T{1,2}=[22 23; 24 25]` on définit la seconde cellule, etc.;
- on peut aussi initialiser le tableau en une seule opération: `T={ 'hello' , [22 23; 24 25]; personne , { 'quatre' , 44 } }` (où on pourrait omettre les virgules).

Il est intéressant de noter que Octave est également capable de sauvegarder des tableaux cellulaires sous forme fichier texte (avec `save -ascii`)… ce que MATLAB ne sait pas faire!

Lecture-écriture de fichiers

Il est souvent nécessaire d'échanger des données entre Octave/MATLAB et d'autres logiciels. Cela s'effectue généralement via des fichiers texte, et l'on utilise le plus souvent les fonctions `save -text` (ou `save -ascii`), `load`, `dlmwrite` et `dlmread`. Moins connues, d'autres fonctions confèrent cependant à Octave et MATLAB la même souplesse d'entrée/sortie que les langages de programmation classiques (C…):

- `textread`: lecture d'un fichier texte dont l'ensemble des données répond à un format homogène;
- `fprintf` et `fscanf`: écriture ou lecture formatée d'un fichier texte;
- `fgetl` et `fgets`: lecture de fichier texte ligne après ligne, resp. lecture d'un nombre de caractères spécifié;
- `fopen`, `feof`, `frewind` et `fclose`: ouverture/fermeture de fichier, et contrôle d'avancement;
- `fwrite` et `fread`: écriture ou lecture binaire.

Éditeurs de code Octave/MATLAB

Octave n'offrant pas en standard un environnement de développement (IDE), il est utile, lorsqu'on édite du code Octave/MATLAB (M-files de scripts ou de fonctions), de disposer d'un éditeur de texte offrant des possibilités de *coloriage syntaxique* pour détecter aisément les fautes de frappe.

Sous Windows, le paquet *stand-alone* actuel Octave 2.1.73 embarque à cet effet l'éditeur Scintilla SciTE[28], mais on peut faire usage d'autres éditeurs de programmation gratuits (p.ex. Crimson Editor[29]), shareware (UltraEdit[30]), etc… On adaptera en conséquence la variable `EDITOR` (chemin d'accès à l'éditeur) dans le script `octave.sh` de démarrage Octave pour pouvoir faire usage, dans la fenêtre Octave, de la commande `edit`.

Dans le monde Linux, les éditeurs de programmation sont légion, et il suffira de choisir celui qui reconnaît le langage MATLAB.

[28] ***http://www.scintilla.org/***
[29] ***http://www.crimsoneditor.com/***
[30] ***http://www.ultraedit.com/***

Écrire du code portable

Pour faciliter le portage de M-files (scripts ou fonctions) de Octave vers MATLAB et vice versa, quelques conseils de programmation peuvent s'avérer utiles. La liste des recommandations qui suit est cependant loin d'être exhaustive.

On s'efforcera de n'utiliser que les fonctions et paramètres communs aux deux environnements. Quelques exemples:

- Sous MATLAB: utiliser les opérateurs `+ - *`… au lieu des fonctions `plus`, `minus`, `mtimes`…; utiliser les opérateurs relationnels `==`, `~=` au lieu des fonctions `eq`, `ne`…; utiliser les opérateurs logiques `~ & |` au lieu des fonctions `not`, `and`, `or`…
- Sous Octave: délimiter les chaînes par des apostrophes et non des guillemets; ne pas utiliser les opérateurs `++` et `--` de pré/post incrémentation/décrémentation à la manière du langage C; déclarer les fonctions et utiliser les structures de contrôle avec la syntaxe MATLAB (pas de `endfor`, `endwhile`…); ne pas utiliser les *listes* Octave mais plutôt les *tableaux cellulaires*.
- Le site de Paul Kienzle[31] (grand contributeur Octave Forge) propose une intéressante *Octave / Matlab Compatibility Database*. Notre cours *Introduction à MATLAB et Octave*[32], qui fait systématiquement le parallèle entre Octave et MATLAB, peut également rendre service dans cette perspective.

Pour des scripts interactifs, on utilisera de préférence les fonctions classiques de dialogue alpha-numérique avec l'utilisateur (`disp`, `fprintf`, `input`, `menu`, `warning`, `error`, `pause`, `waitbar`…) plutôt que les fonctionnalités MATLAB sophistiquées d'élaboration de GUI (interface utilisateur graphique) qui n'existent pas en standard (ou pas sous la même forme) dans Octave.

Si l'on souhaite réaliser du code contenant des parties spécifiques à Octave et à MATLAB, on peut élaborer du *code conditionnel* testant l'environnement dans lequel le script s'exécute avec:

```
if ~ exist('OCTAVE_VERSION')
  % ici instruction(s) pour MATLAB
else
  % ici instruction(s) équivalente(s)
  % pour Octave
end
```

On rappelle encore que la principale cause d'incompatibilité de scripts entre Octave et MATLAB provient des fonctions graphiques (plus étendues sous MATLAB qu'avec le couple Octave-Gnuplot). La solution réside, pour l'instant, dans l'usage de grapheurs alternatifs (Octaviz, Octplot…) tout en faisant du *code conditionnel* garantissant la portabilité.

Conclusion

Octave, logiciel libre et projet très vivant de la communauté Internet, va certainement continuer d'évoluer. Il n'atteindra cependant jamais l'étendue fonctionnelle de MATLAB qui, grâce à ses innombrables *toolboxes*, restera longtemps présent dans les diverses disciplines où il connaît du succès.

Nous avons cependant constaté ces dernières années, au fil de nos expériences avec les étudiants ainsi que dans nos contacts avec d'autres enseignants, que Octave suffit en général largement dans la plupart des enseignements qui font traditionnellement appel à MATLAB (analyse numérique, statistiques, introduction à la modélisation…).

Octave présente, de notre point de vue, au moins trois atouts importants (qui sont du reste l'apanage de tous les logiciels libres):

- il n'occasionne aucun coût (pas de frais de licence et de mises à jour), qu'il soit distribué et utilisé sur des postes de l'École (laboratoires, salles d'enseignement…) ou sur des machines privées (collaborateurs, étudiants);
- il est disponible sur les trois plates-formes majeures que sont Windows, Linux et Mac OS X;
- la possibilité de contribuer activement à son développement est particulièrement intéressante et motivante dans le monde académique où liberté et créativité restent des valeurs primordiales.

Références Internet

Sites de base

- Site GNU Octave:
  http://www.gnu.org/software/octave/
- Site Octave-Forge (GNU Octave Repository):
  http://octave.sourceforge.net/

Installer et utiliser Octave

Conseils sur l'installation de Octave-Forge

- sous Windows:
  http://wiki.octave.org/wiki.pl?OctaveForWindows et
  http://enacit1.epfl.ch/cours_matlab/octave.html
- sous Linux:
  http://wiki.octave.org/wiki.pl?OctaveForGNULinux
- sous MacOS X:
  http://wiki.octave.org/wiki.pl?OctaveForMac

Téléchargement

- GNU Octave (sources du noyau de base):
  http://www.gnu.org/software/octave/download.html
- Octave-Forge (paquetages, sources, CVS…):
  http://sourceforge.net/project/showfiles.php?group_id=2888

Introduction à Octave

- cours avec comparatif Octave vs. MATLAB:
  http://enacit1.epfl.ch/cours_matlab/
- manuel GNU Octave (sans ext. Octave-Forge):
  http://www.gnu.org/software/octave/doc/interpreter/
- supplément à ce manuel:
  http://wiki.octave.org/wiki.pl?CategoryManual
- manuel des fonctions Octave-Forge:
  http://octave.sourceforge.net/index/
- FAQ Octave:
  http://www.gnu.org/software/octave/FAQ.html
- description de Octave sur Wikipedia:
  http://en.wikipedia.org/wiki/GNU_Octave

[31] ***http://users.powernet.co.uk/kienzle/octave/matcompat/***
[32] ***http://enacit1.epfl.ch/cours_matlab/***

Octave *versus* MATLAB

- *compatibility FAQ* : ***http://octave.sourceforge.net/compatibility.html***
- *compatibility Database* : ***http://users.powernet.co.uk/kienzle/octave/matcompat/***
- *compatibility list* du Wiki Octave: ***http://wiki.octave.org/wiki.pl?MatlabOctaveCompatibility***

- projets de développement de paquetages: ***http://wiki.octave.org/wiki.pl?CategoryCode***

Les principaux *back-ends* graphiques pour Octave

- Gnuplot: ***http://gnuplot.sourceforge.net/***
- Octaviz: ***http://octaviz.sourceforge.net/***
- OctPlot: ***http://octplot.sourceforge.net/***
- epsTK: ***http://www.epstk.de/***

Communiquer autour d'Octave… et contribuer au projet !

- *Mailing-list* sur Octave (souscription, accès archives): ***http://www.gnu.org/software/octave/archive.html***
- Wiki officiel sur Octave: ***http://wiki.octave.org/***
- Autre wiki, orienté utilisation de Octave: ***http://www.aims.ac.za/wiki/index.php/Octave***
- Blog des ***maintainers@octave.org***: ***http://blog.gmane.org/gmane.comp.gnu.octave.maintainers/***
- Forum SourceForge relatif à Octave: ***http://sourceforge.net/forum/?group_id=2888***
- *Channel IRC* : ***http://wiki.octave.org/wiki.pl?OctaveIRC***
- *Developer's Guide* (pour contribuer au projet): ***http://octave.sourceforge.net/new_developer.html***

Scripts et fonctions

- Sur le site de *The MathWorks* (bon nombre des contributions téléchargeables tournent sur Octave, ou ne nécessitent que de légères adaptations !): ***http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do*** ■